4th Planetary Data Workshop

An Approach Towards Supporting Seamless Search Across PDS3 and PDS4 Metadata

Kevin Grimes – kevin.m.grimes@jpl.nasa.gov

Co-Authors: Anna Waldron, Rishi Verma, Cristina DeCesare, Paul Ramirez, Jordan Padams, Sean Hardman, Michael Cayanan

Flagstaff, Arizona

Wednesday, June 19, 2019

Jet Propulsion Laboratory
California Institute of Technology

Seamless Search Across PDS3 and PDS4 Metadata

Overview

- Introduction
- Cross-Standard Search
- Increased Process Automation
- Further Work
- Questions and References

Introduction

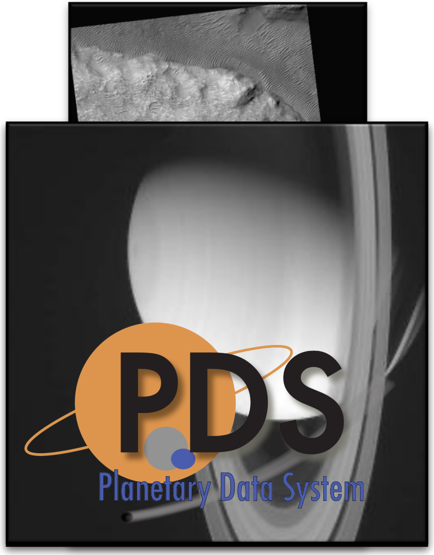
Overview

- PDS Imaging Node
- Metadata Standards
  - PDS3
  - PDS4
- Search Challenges

Introduction

PDS Imaging Node

- Cartography and Imaging Sciences Node (IMG) of the NASA Planetary Data System (PDS)
- Home to over 1380 TB[1] of digital image archives
- Diverse collection of images
  - Both orbital and landed missions
    - Over 20 million images taken from the surface of Mars
    - Nearly 5 million images taken of Mars's surface from orbit
    - Images of Jupiter, Saturn, and beyond
  - Original, raw experiment data and derived products
  - Differing coordinate systems



Introduction

PDS Imaging Node

- Photojournal
  - Primary interface to the Planetary Image Archive (PIA)
  - Public-facing, thousands of high-resolution images
  - https://photojournal.jpl.nasa.gov
- Image Atlas
  - Search by PDS label content and additional metadata
  - Powered by Apache Solr, open API
  - https://pds-imaging.jpl.nasa.gov/search

Introduction

Overview

- PDS Imaging Node
- Metadata Standards
  - PDS3
  - PDS4
- Search Challenges

Introduction

Metadata Standards – PDS3

- Key-value pairs
- Some support for data structures
- Readability
  - Difficult for machines to read
  - Trivial for humans to understand
- Tooling support
  - GDAL[2]
  - ISIS[3]
  - VICAR[4]

```
/home/kgrimes/shared/demo_images/quick_test/ChemCam > $R2LIB/label
-list FLA_349859878EDR_F0000000001036288Z1.VIC
Beginning VICAR task LABEL
LABEL version 2019-05-28
************************************************************

        ************  File
FLA_349859878EDR_F0000000001036288Z1.VIC ************
                3 dimensional IMAGE file
                File organization is BSQ
                Pixels are in HALF format from a X86-LINUX host
                1 bands
                1024 lines per band
                1024 samples per line
                0 lines of binary header
                0 bytes of binary prefix per line
---- Property: IDENTIFICATION ----
DATA_SET_ID='SIM-M-HAZCAM-2-EDR-OPS-V1.0'
DATA_SET_NAME=
'SIMULATED MARS SCIENCE LABORATORY HAZARD AVOIDANCE CAMERA EDR OPS
VERSION 1.0'
COMMAND_SEQUENCE_NUMBER=0
FRAME_ID='LEFT'
FRAME_TYPE='STEREO'
GEOMETRY_PROJECTION_TYPE='RAW'
IMAGE_ID='25'
IMAGE_TYPE='REGULAR'
IMAGE_ACQUIRE_MODE='IMAGE'
INSTRUMENT_HOST_ID='SIM'
INSTRUMENT_HOST_NAME='SIMULATED MARS SCIENCE LABORATORY'
INSTRUMENT_ID='FRONT_HAZCAM_LEFT_A'
INSTRUMENT_NAME='FRONT HAZARD AVOIDANCE CAMERA LEFT A'
INSTRUMENT_SERIAL_NUMBER=27
INSTRUMENT_TYPE='IMAGING CAMERA'
INSTRUMENT_VERSION_ID='BB'
```

Introduction

Metadata Standards – PDS4

```xml
<Identification_Area>
  <logical_identifier>urn:nasa:pds:insight_cameras:data:c000m0001_596620131
  <version_id>5.0</version_id>
  <title>InSight ICC EDR  Observational Product - c000m0001_596620131edr_f0
  <information_model_version>1.11.1.0</information_model_version>
  <product_class>Product_Observational</product_class>
  <Alias_List>
    <Alias>
      <alternate_id>C000M0001_596620131EDR_F0000_0589M5</alternate_id>
      <comment>VICAR PRODUCT_ID</comment>
    </Alias>
  </Alias_List>
</Identification_Area>
<Observation_Area>
  <comment>Observational Intent</comment>
  <Time_Coordinates>
    <start_date_time>2018-11-27T19:49:16.116Z</start_date_time>
    <stop_date_time>2018-11-27T19:49:16.445Z</stop_date_time>
    <local_mean_solar_time>13:36:23.560</local_mean_solar_time>
    <local_true_solar_time>12:58:09</local_true_solar_time>
    <solar_longitude unit="deg">296.258</solar_longitude>
  </Time_Coordinates>
  <Primary_Result_Summary>
    <purpose>Science</purpose>
    <processing_level>Raw</processing_level>
    <Science_Facets>
      <wavelength_range>Visible</wavelength_range>
      <domain>Surface</domain>
```

- XML-formatted
- Forced compliance to PDS schema[5]
- Readability
  - Trivial for machines to read
  - Difficult for humans to understand at first glance
- Tooling support
  - A lot

Introduction

Overview

- PDS Imaging Node
- Metadata Standards
  - PDS3
  - PDS4
- Search Challenges

Introduction

Search Challenges

- Determining equivalence between a given PDS3 keyword and a PDS4 X-Path
  - PDS3: `IDENTIFICATION.TARGET_NAME`
  - PDS4: `//pds:Target_Identification/pds:name`
- X-Paths as search keywords is cumbersome

Seamless Search Across PDS3 and PDS4 Metadata

Overview

- Introduction
- Cross-Standard Search
- Increased Process Automation
- Conclusions
- Questions and References

Cross-Standard Search

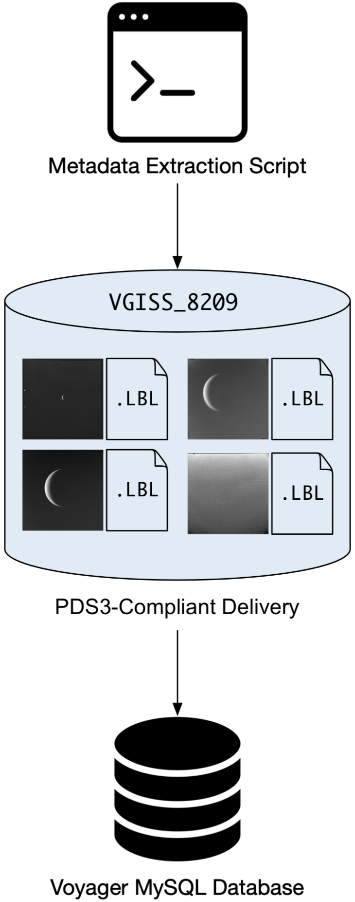
Overview

- Existing Architecture
- Motivation to Upgrade
- Updated Data Ingestion Model
    - Metadata Extraction
        - Label Mapping Tool
        - Harvest
        - Search Service
    - Searching with Solr
        - Collection Sharding
        - Index Update Procedure

Cross-Standard Search

Existing Architecture

- Mission-specific metadata extraction scripts
  - Loop through each image label
  - Parse out relevant metadata into MySQL database
  - New set of scripts for every mission, instrument, and product type combination
- Requires low-level technical understanding to debug
- Supports PDS3-formatted labels only



Cross-Standard Search

Existing Architecture

- Solr configuration
  - Edit `schema.xml` file to include new fields being parsed out of the labels
  - Manually reload Solr configuration
- Update Solr index
  - Solr queries relevant MySQL database for new records
  - May take several hours
- Solr specifications
  - 32 GB RAM
  - Total index size: about 80 GB
  - Single copy of the index

Cross-Standard Search

Overview

- Existing Architecture
- Motivation to Upgrade
- Updated Data Ingestion Model
  - Metadata Extraction
    - Label Mapping Tool
    - Harvest
    - Search Service
  - Searching with Solr
    - Collection Sharding
    - Index Update Procedure

Cross-Standard Search

Motivation to Upgrade

- Metadata extraction scripts
  - Per mission/instrument/product
  - Usually written by interns
  - Prone to break
  - Little error handling
  - No PDS4 support
- Solr infrastructure
  - Entire index on single Solr core
  - No redundancy
  - Semi-frequent crashing
- Three copies of the metadata
  - In the labels themselves
  - In the MySQL database
  - In the Solr index
- Large amounts of manual effort to complete release
  - Humans need to check each piece of the process before continuing
  - Typos can ruin hours of effort

Cross-Standard Search

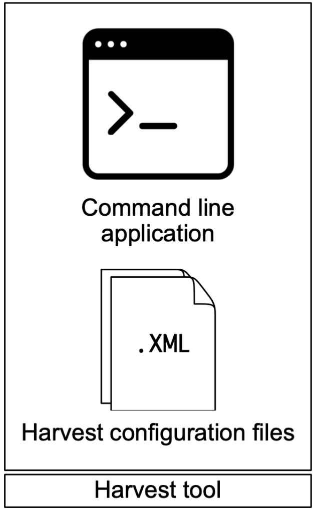
Overview

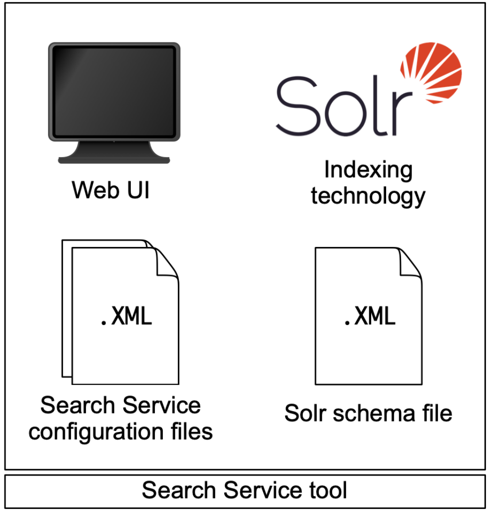
- Existing Architecture
- Motivation to Upgrade
- Updated Data Ingestion Model
  - Metadata Extraction
    - Label Mapping Tool
    - Harvest
    - Search Service
  - Searching with Solr
    - Collection Sharding
    - Index Update Procedure

Cross-Standard Search

Updated Data Ingestion Model – Metadata Extraction

- Replace sets of scripts with PDS Engineering Node[6] tools
    - Harvest[7]
    - Search Service[8]
- Some assembly required
    - Several configuration files
    - Setup and teardown





Cross-Standard Search

Metadata Extraction – Label Mapping Tool

- Maintains knowledge of which PDS3 keywords map to which PDS4 X-Paths
  - Support for one-to-one, one-to-many, and many-to-many relationships
  - Stores *common names*: logical, unofficial synonyms for entries
    - PDS3: `IDENTIFICATION.TARGET_NAME`
    - PDS4: `//pds:Target_Identification/pds:name`
    - Common name: `target`
- Initial ingestion from Imaging Ingest Local Data Dictionary (IILDD)
- Queries and further updates can be made via RESTful API
- *Currently* internal to JPL only

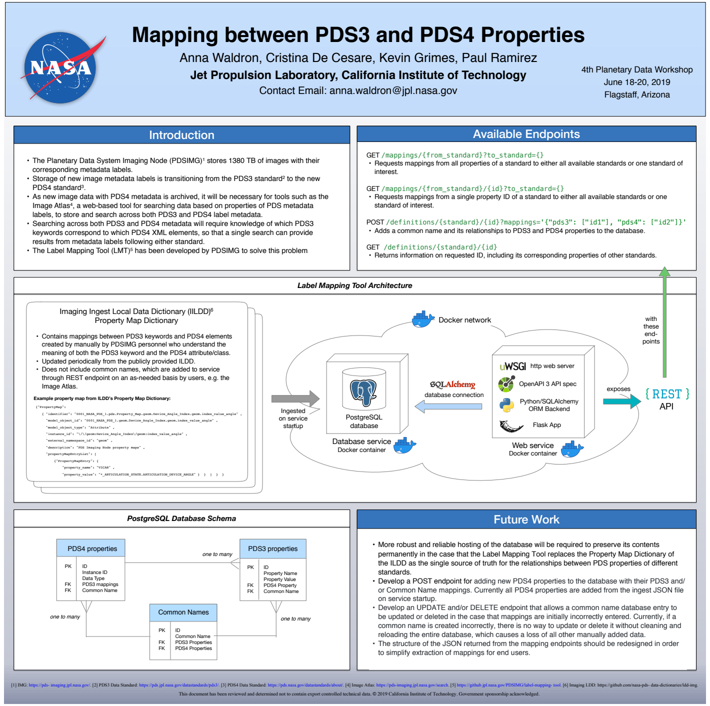
Cross-Standard Search

Metadata Extraction – Label Mapping Tool

Shameless plug:

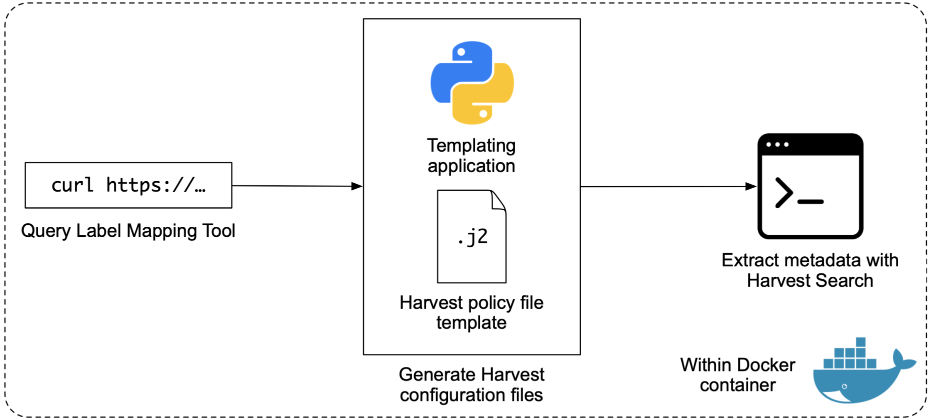
Check out “Mapping between PDS3 and PDS4 Properties” by Anna Waldron, *et al.* during the poster session for more info on the Label Mapping Tool!

- Example usage
- Database design
- Technologies



Cross-Standard Search

Metadata Extraction – Harvest



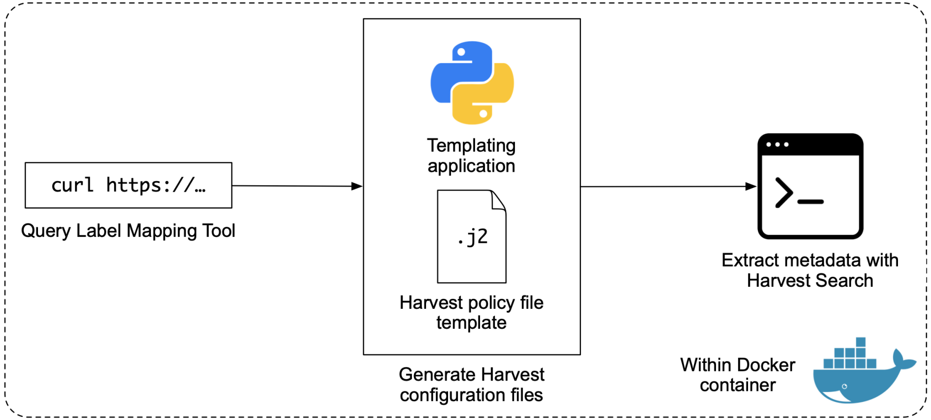
Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Harvest

1. Query Label Mapping Tool for latest PDS3/PDS4/common name mappings

Cross-Standard Search

Metadata Extraction – Harvest



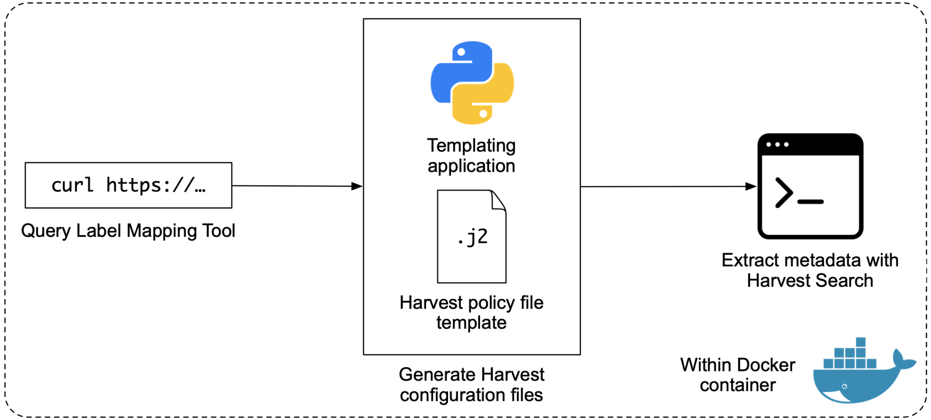
Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Harvest

2. Generate Harvest policy file with keywords to extract from PDS4 bundle, and their common name equivalents (“slot names”)

Cross-Standard Search

Metadata Extraction – Harvest



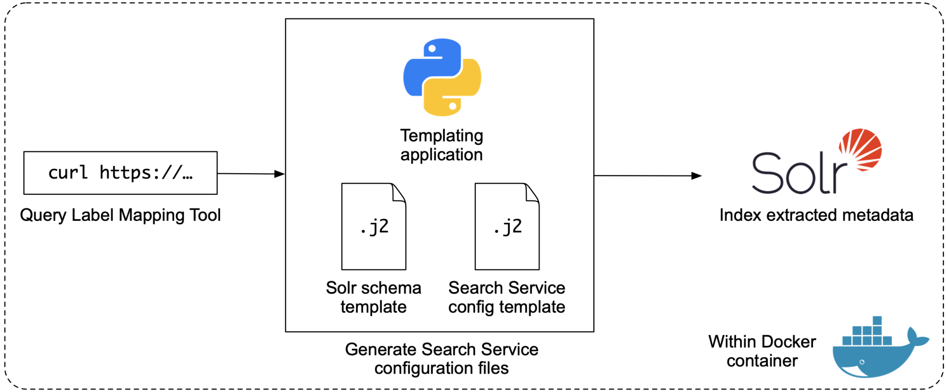
Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Harvest

3. Run Harvest
4. Store results
5. Destroy container

Cross-Standard Search

Metadata Extraction – Search Service



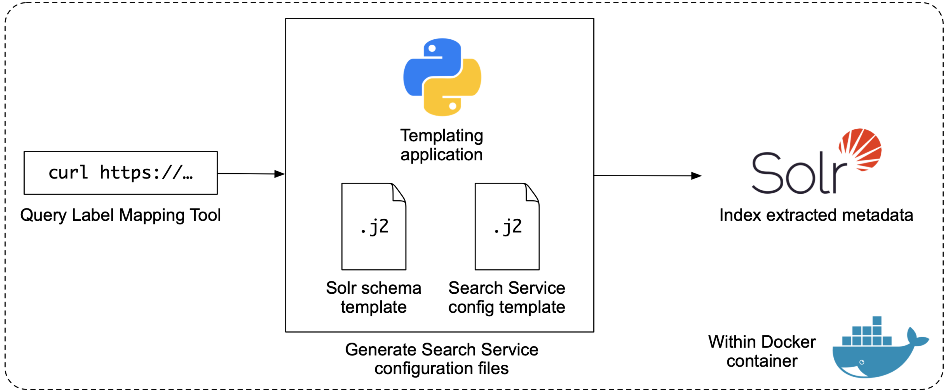
Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Search Service (including Solr)

1. Query Label Mapping Tool for latest PDS3/PDS4/common name mappings

Cross-Standard Search

Metadata Extraction – Search Service



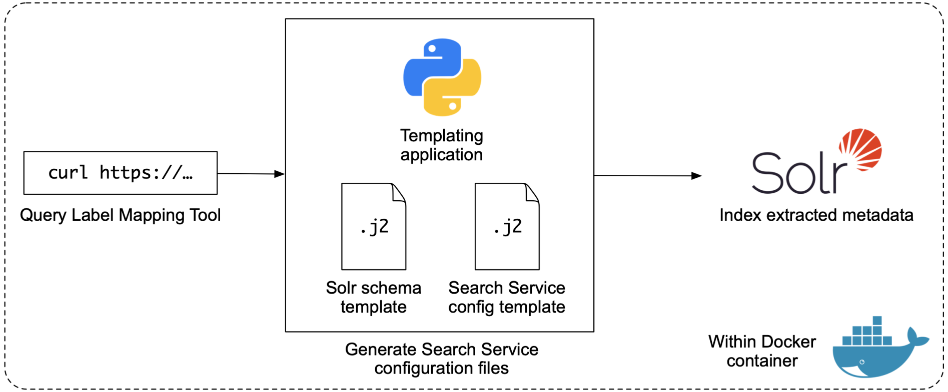
Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Search Service (including Solr)

2. Generate Solr schema, with common names ("slot names") as fields
3. Generate Search Service config file

Cross-Standard Search

Metadata Extraction – Search Service



Upgraded procedure leverages several technologies:

- Python + Jinja2 templating engine [6]
- Label Mapping Tool
- Docker
- Search Service (including Solr)

4. Index metadata extracted by Harvest
5. Dump index to disk
6. Destroy container

Cross-Standard Search

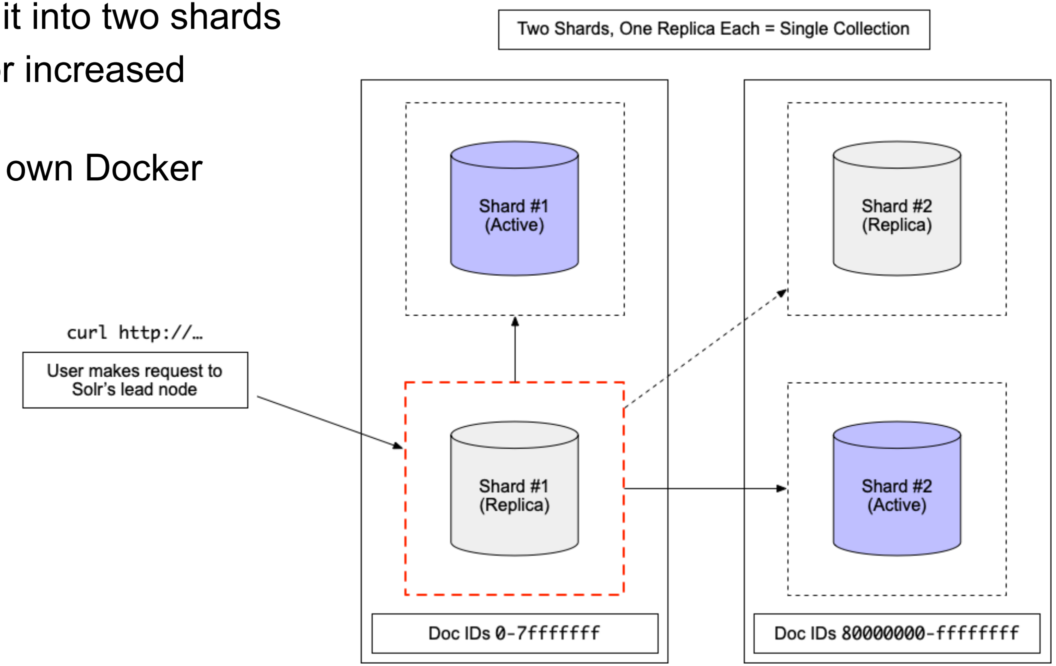
Overview

- Existing Architecture
- Motivation to Upgrade
- Updated Data Ingestion Model
    - Metadata Extraction
        - Label Mapping Tool
        - Harvest
        - Search Service
    - Searching with Solr
        - Collection Sharding
        - Index Update Procedure

Cross-Standard Search

Updated Data Ingestion Model – Programmatic Schema Updates

- Collection sharding
  - Single `pds_archives` collection split into two shards
  - Each Solr shard replicated once for increased availability
  - Each shard replica served from its own Docker container



Cross-Standard Search

Updated Data Ingestion Model – Programmatic Schema Updates

- Index update procedure
  - Solr schema managed by Zookeeper
  - Simple schema updates made using Solr's RESTful API
  - Field representation
    - PDS3 keyword
    - PDS4 X-Path
    - Common names

```
<schema>
  ...
  <field name="//pds:Investigation_Area/pds:name" type="text_general_facet" indexed="true" stored="true" />
  <field name="INVESTIGATION.MISSION_NAME" type="text_general_facet" indexed="true" stored="true" />
  ...
  <copyField source="//pds:Investigation_Area/pds:name" dest=”mission" />
  <copyField source="INVESTIGATION.MISSION_NAME" dest="mission” />
  ...
</schema>
```

Seamless Search Across PDS3 and PDS4 Metadata

Overview

- Introduction
- Cross-Standard Search
- Increased Process Automation
- Conclusions and Further Work
- Questions and References

Increased Process Automation

Overview

- Existing Procedures
  - Data Release
  - Image Atlas Ingestion
- Motivation to Upgrade
- Upgraded Procedures
  - Ansible
  - XL Release

Increased Process Automation

Existing Procedures

- Data Release
  - Data provider alerts IMG that their data is ready
  - Delivery process is instantiated, either via Internet transfer or snail mail
  - IMG validates delivery and generates checksums
  - Data is moved to archive on release day
- Image Atlas Ingestion
  - Operations lead is notified that data has been successfully archived
  - Lead runs several scripts on data, extracting the metadata and ingesting it into Solr
  - Blue-green switch is flipped, making the updated Solr index live

Increased Process Automation

Overview

- Existing Procedures
  - Data Release
  - Image Atlas Ingestion
- Motivation to Upgrade
- Upgraded Procedures
  - Ansible
  - XL Release

Increased Process Automation

Motivation to Upgrade

- Too many people in the loop!
  - Data provider
  - IMG release lead
  - IMG operations lead
- Errors are difficult to remedy
  - Requires low-level process knowledge
  - Substantial effort to undo changes thus far
  - Little to no logging
- Several manual steps
  - Few actions can be CRONed
  - Multiple scripts
- The copy being worked on is *the only copy*
  - Image Atlas
  - Data websites

Increased Process Automation

Overview

- Existing Procedures
  - Data Release
  - Image Atlas Ingestion
- Motivation to Upgrade
- Upgraded Procedures
  - Ansible
  - XL Release

Increased Process Automation

Upgraded Procedures

- Ansible: "Configuration Management for developers"
- Ansible *playbooks*
    - Series of *roles* to be executed on machines
    - Roles are accomplished by *tasks*
    - Parts of playbooks may be run with *tags*
- Vast selection of *modules* can reduce entire scripts to four or five lines

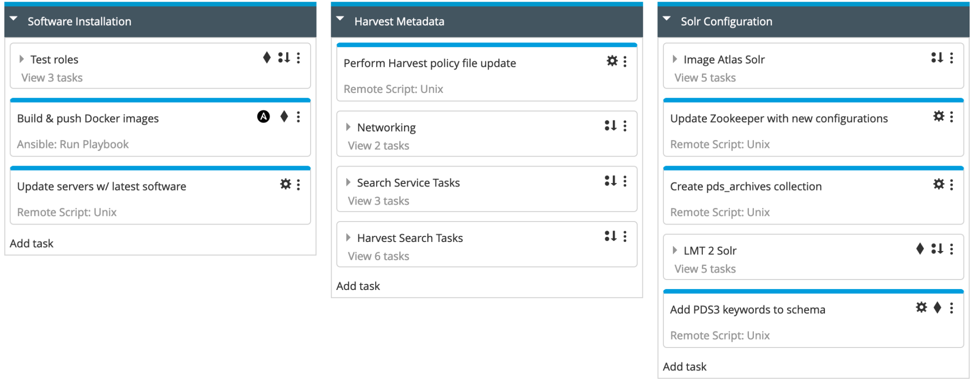
```
- name: "deploy and unpack search-core tarball"
  unarchive:
    src: "search-core-{{ search_core.version }}-bin.tar.gz"
    dest: "{{ task_work_dir }}"
    creates: "{{ task_work_dir }}/search-core-\
      {{ search_core.version }}"
    keep_newer: true
  tags:
    - "ia-ingestion-pipelines"
    - "ia-ingestion-pipelines-insight"
    - "ia-ingestion-pipelines-insight-extract-metadata"
    - "ia-ingestion-pipelines-insight-extract-metadata-deploy"
    - "search-service"
    - "search-core"

- name: "put search service schema into place"
  template:
    src: "search-service-schema.xml.j2"
    dest: "{{ task_work_dir }}/search-service-{{ search_service.version }}/\
      pds/conf/schema.xml"
  tags:
    - "ia-ingestion-pipelines"
    - "ia-ingestion-pipelines-insight"
    - "ia-ingestion-pipelines-insight-extract-metadata"
    - "ia-ingestion-pipelines-insight-extract-metadata-deploy"
    - "search-service"
```

Increased Process Automation

Upgraded Procedures

- XL Release
  - "Application Release Orchestration"
  - But it works for data releases, too!
- Web interface
- Easy pipeline design
- Reusable templates
- Invite others to complete tasks
  - Data providers
  - Operations lead approval



Seamless Search Across PDS3 and PDS4 Metadata

Overview

- Introduction
- Cross-Standard Search
- Increased Process Automation
- Further Work
- Questions and References

Further Work

- Migrate from XL Release to Ansible Tower or AWX
- Use Kubernetes + Rancher to orchestrate container provisioning
- Migrate existing PDS3 processes to use Ansible + XLR
- Replace Image Atlas Solr with Search Service
  - Both powered by Solr
  - Search Service has some extra functionality

Seamless Search Across PDS3 and PDS4 Metadata

Overview

- Introduction
- Cross-Standard Search
- Increased Process Automation
- Further Work
- Questions and References

Questions and References

Questions?

[1] As of May 2019.
[2] GDAL: https://gdal.org/
[3] ISIS: https://isis.astrogeology.usgs.gov/
[4] VICAR: https://www-mipl.jpl.nasa.gov/external/vicar.html
[5] PDS Schema: http://pds.nasa.gov/pds4/pds/v1
[6] Jinja2: http://jinja.pocoo.org/docs/2.10/

Slide 13 graphics obtained from Stenciltown. Property of The Omni Group.
Slide 14, 18, 24, 25, 26, 28, 29 "Solr Logo" property of The Apache Foundation.

Slide 21, 22, 23, 24 "Python Logo" property of The Python Foundation.
Slide 21, 22, 23, 24 "Docker Logo" property of Docker, Inc.
Slide 29 "Zookeeper Logo" property of The Apache Foundation.
Slide 36 "Ansible Logo" property of Redhat, Inc.
Slide 37 "XL Release Logo" property of Xebia Labs.
Slide 39 "Red Hat Ansible Tower Logo" property of Redhat, Inc.
Slide 39 "Rancher Logo" property of Rancher Labs.
Slide 39 "Kubernetes Logo" property of The Linux Foundation.

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov